Classification of Urdu News Articles

Jibran Mazhar
Rayan Atif
Rahimah Siddiqi
Ibrahim Murtaza
Hadia Faisal

ABSTRACT

This paper provides a classification algorithm for categorizing Urdu news articles into five categories: Entertainment, Business, Sports, Science-Technology, and International. Three models, Multinomial Bayes, Logistic Regression, and Simple Neural Network, are compared on a cleaned dataset using Bag Of Words for feature extraction. This data (in the form of articles) was scraped from relevant categories of three Urdu news websites: Geo Urdu, Jang, and Express. It was then preprocessed to remove duplicate values, missing values, and less informative articles to ensure high-quality input. While all three models performed similarly on 1139 articles, Neural Networks remained ahead with 97.8% accuracy compared to Multinomial Naive Bayes with 95.59% and Logistic Regression with 95.61%. This shows that the ability of Neural Networks to capture non-linear and often complex patterns proves helpful in classifying Urdu text. Hence, this work contributes to improving AI-based content categorization for Urdu, with the potential to extend it to other languages and take it beyond just news articles and into more domains.

ACM Reference Format:
Jibran Mazhar, Rayan Atif, Rahimah Siddiqi, Ibrahim Murtaza, and Hadia Faisal. 2024. Classification of Urdu News Articles. In . ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 METHODOLOGY

After identifying websites for Urdu news articles, data scraping was carried out on each using Python's Beautiful Soup and requests libraries, which were then compiled as a CSV file. Relevant fields in each website were extracted by parsing their HTML structure and locating content areas/main bodies of articles.

With 1139 articles attained from scraping, the CSV file was analyzed and contained rows where content and/or title fields were missing. As a result, these rows, along with rows containing duplicate articles, were dropped. Moreover, missing values in columns other than "title" and "content" were filled with placeholder values. Finally, the resulting cleaned data was converted into tokens and, from that, Bag of Words.

Logistic Regression and Multinomial Bayes models were implemented with an 80/20 split for training and testing data, respectively.

The Multinomial Bayes model was equipped with an additional layer of preprocessing by converting tokens into n-grams (which, after tuning, were reduced to bigrams). By using n-grams, we are able to add context to the individual tokens, capturing relationships between consecutive words. This helps the model better understand word dependencies and semantic meaning, allowing for more accurate predictions, especially in our case, where context plays a significant role in classifying or predicting outcomes. The model was implemented using Lidstone Smoothing (instead of Laplace Smoothing) with an alpha value of 0.01 chosen after hyperparameter tuning. Predictions were made on test data after fitting training data by calculating the probability of each row in test data belonging to a specific class, with the highest probability class being selected.

For Logistic Regression, we implement One vs All classification using a tokenized Bag of Words and Cross Entropy Loss, which is well-suited for classification tasks involving multiple classes. This loss function measures the discrepancy between the predicted probability distribution and the true distribution, effectively guiding the model to improve its predictions. The Bag Of Words was made by combining individual Bags of Words of "Title" and "Content", and this was used as input to the model.

The neural network architecture consisted of an input layer whose size matched the dimensionality of the combined feature vectors. This was followed by a hidden layer comprising 256 neurons, utilizing the Rectified Linear Unit (ReLU) activation function. The choice of ReLU was motivated by its ability to introduce non-linearity to the model while mitigating issues like the vanishing gradient problem, thus facilitating faster and more efficient training. The output layer contained neurons equal to the number of target categories, employing the softmax activation function to produce a probability distribution over the classes. The softmax function was essential for multi-class classification as it ensured that the output probabilities were summed to one, allowing for a precise prediction of the most likely category for each article.

Along with Cross Entropy Loss, the Adam optimizer was chosen to update the network weights during training due to its adaptive learning rate capabilities, which combine the advantages of both AdaGrad and RMSProp optimizers. This allowed for efficient and stable convergence of the model parameters.

The training process involved iteratively feeding the training data through the network and adjusting the weights to minimize the loss function. The validation set was used to monitor the model's performance and prevent overfitting by tuning hyperparameters and implementing early stopping if necessary. After training, the model's effectiveness was evaluated on the testing set, ensuring the

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

results reflected its ability to generalize to unseen data. To further confirm model correctness, a sample of 20 randomly selected news article titles, along with their predicted and true categories, were printed.

2 FINDINGS

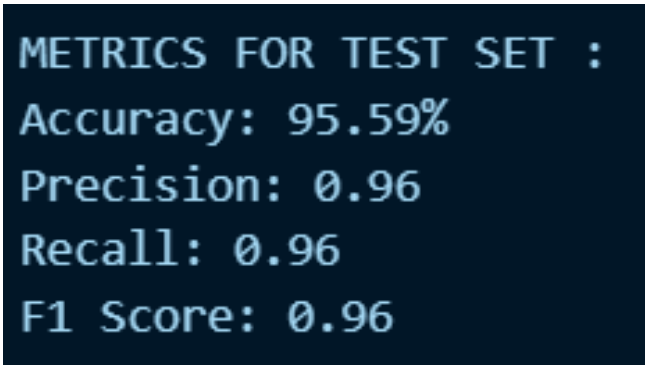

Figure 1: Multinomial Bayes Results

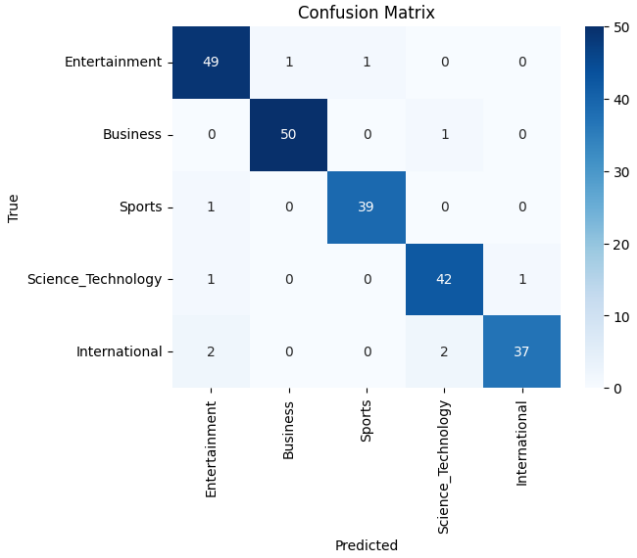

Figure 2: Multinomial Bayes Confusion Matrix

Figure 1 shows Multinomial Bayes resulting in an accuracy of 95.59%. Since MNB uses probabilities to predict whether an article belongs to a certain class based on the frequency of words (bigrams in this case), it is well-suited for this text data. This is shown by its high accuracy and high f1 score, indicating that it predicts correctly while also appropriately handling false positives and false negatives. The confusion matrix in Figure 2 supports this conclusion, as the majority of results lie in the diagonal of the matrix.

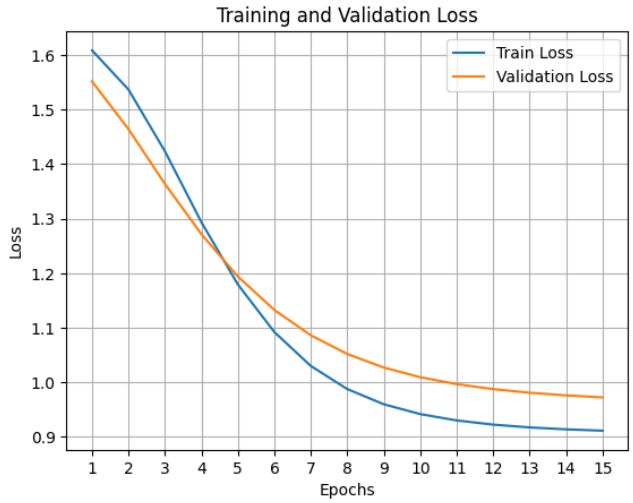

Figure 3: Neural Network Loss

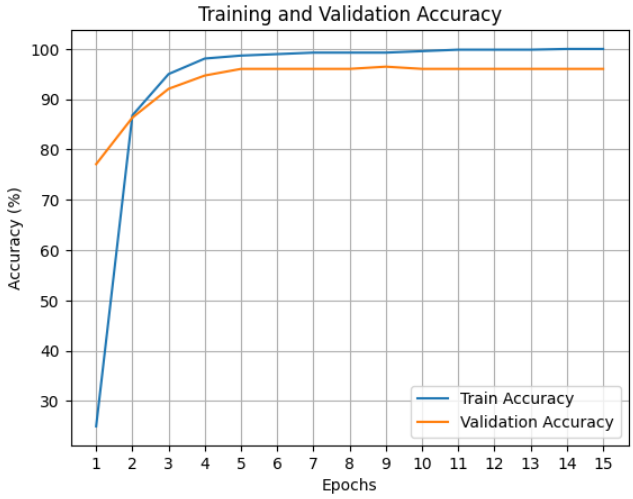

Figure 4: Neural Network Accuracy

Figures 3 and 4 show Neural Network Loss and Accuracy, respectively. Both validation loss and training loss, as well as training accuracy and validation accuracy, follow the same trend. This shows no overfitting or underfitting since loss/accuracy for either one would have started increasing/decreasing accordingly. Since both loss and accuracy curves become stable, we can conclude that the model achieved good generalization.

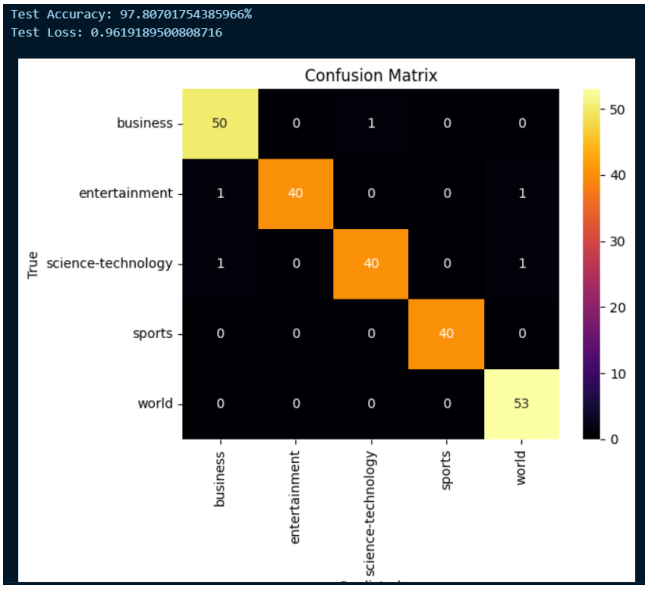

Figure 5: Neural Network Results

```
Label Mapping: {'entertainment': 0, 'business': 1, 'sports': 2, 'science-technology': 3, 'world': 4}
Training classifier for category: 0
Final loss for category 0: 0.01489320215243476
Training classifier for category: 1
Final loss for category 1: 0.0129799277052144 58
Training classifier for category: 2
Final loss for category 2: 0.014559640359764214
Training classifier for category: 3
Final loss for category 3: 0.01894340798801794
Training classifier for category: 4
Final loss for category 4: 0.022747184803912 4
Accuracy: 0.956140350877193
F1 Score: 0.9558654475129876
Confusion Matrix:
[[42  0  0  0  1]
 [ 0 47  0  4  3]
 [ 0  0 34  0  0]
 [ 0  0  0 52  0]
 [ 1  0  0  1 43]]
```

Figure 6: Logistic Regression Results

Figure 6 represents Logistic Regressions results. The results shown are very similar to those of Multinomial Bayes, with an accuracy of 95.61% and an f-1 Score of 95.59%. This is primarily because both Multinomial Bayes and Logistic Regression are linear classifiers, which means they both aim to find a linear decision boundary in the feature space. Even though Multinomial Bayes is based on a probabilistic model and Logistic Regression relies on a log-odds approach, both methods are capable of producing similar decision boundaries when the data is linearly separable. Additionally, the feature sets, despite being different (unigrams in Logistic Regression vs. bigrams in Multinomial Bayes), likely contain enough overlapping information for both models to perform similarly, especially if the dataset has clear and distinct patterns that both models can exploit. This is further supported by well-tuned hyperparameters in both models, ensuring that they both generalize effectively and perform well on the test data.

However, as can be seen in Figure 5, Neural Network results in the highest accuracy among the 3 models with 97.8%. This is largely due to Neural Networks being able to learn complex patterns along with non-linear relationships, which are common in language texts, especially Urdu, that Multinomial Bayes and Logistic Regression may miss since they assume data to have a linear relationship. As a result, Neural Networks are able to learn complex linear boundaries and generalize to unseen data better than the other 2, resulting in a higher accuracy.

3 LIMITATIONS AND CONCLUSION

The model developed in this study is based on a dataset of 1139 Urdu news articles, which, while substantial, may not be large enough to ensure optimal generalization across a broader range of Urdu text. A larger and more diverse dataset, incorporating articles from various sources and genres such as blogs, social media, and literature, would likely improve the model's ability to handle the diverse linguistic nuances present in Urdu. Additionally, Urdu is a morphologically rich language, and the preprocessing steps, including tokenization and stemming, may not fully capture the complexity of the language. Words in Urdu often have multiple meanings or contextual uses, which may not be effectively accounted for by the current feature extraction method, limiting the model's accuracy.

The Bag of Words technique, used for feature extraction in this study, has its limitations, especially when working with languages like Urdu, where word order and context are crucial for understanding meaning. While Bag of Words is a common and effective method for many text classification tasks, more advanced techniques such as word embeddings (e.g., Word2Vec or GloVe) or transformer-based models (e.g., BERT for Urdu) could better capture the semantic relationships between words and improve model performance. Furthermore, while Neural Networks achieved the highest accuracy, they come with increased computational complexity and longer training times compared to simpler models like Multinomial Naive Bayes and Logistic Regression. In environments where speed and efficiency are more important than accuracy, these simpler models offer viable alternatives, despite their limitations in capturing complex patterns.

The generalization of this model to other domains beyond news articles, such as product reviews or social media content, is another area that requires attention. The vocabulary and writing styles in different domains can vary significantly, potentially affecting the model's performance. Domain adaptation would be necessary to improve the model's applicability across various contexts.

In conclusion, this paper highlights the effectiveness of different machine learning models for classifying Urdu news articles into predefined categories. Neural Networks, with their ability to capture complex patterns and non-linear relationships, achieved the highest accuracy, but their computational demands may not make them suitable for all applications. On the other hand, Multinomial Naive Bayes and Logistic Regression, despite being simpler models that assume linear relationships, performed similarly well and may be more efficient in scenarios where computational resources are limited.